



[12] 发明专利说明书

专利号 ZL 200510034336.0

[45] 授权公告日 2009年2月4日

[11] 授权公告号 CN 100459717C

[22] 申请日 2005.4.20

[21] 申请号 200510034336.0

[73] 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

[72] 发明人 罗忠 宋彬 常义林 周宁兆

[56] 参考文献

CN1307430A 2001.8.8

CN1466388A 2004.1.7

US6421387B1 2002.7.16

CN1554151A 2004.12.8

H.264/AVC 视频编码标准的技术特点和评价. 毕厚杰. 当代通信, 第8期. 2004

基于 H.26L 的视频抗误码方法研究. 宋彬, 常义林. 西安电子科技大学学报(自然科学版), 第30卷第4期. 2003

视频通信中的误码掩盖技术初探. 龚时鏐, 朱秀昌. 电视技术, 第216期. 2000

审查员 李熙

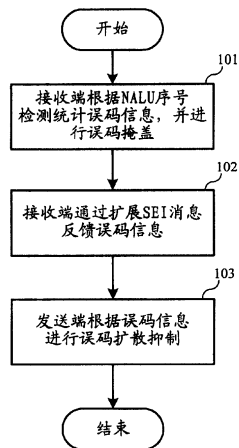
权利要求书 2 页 说明书 15 页 附图 2 页

[54] 发明名称

基于 H.264 的压缩视频传输误码消除方法

[57] 摘要

本发明涉及压缩视频传输误码消除方法, 公开了一种基于 H.264 的压缩视频传输误码消除方法, 使得可以简单高效地消除误码。本发明中, 通过误码信息反馈机制结合误码掩盖和误码扩散抑制实现误码消除; 通过 NALU 序号及承载 Slice 的信息检测误码发生, 并统计丢失数据位置等误码信息; 通过定义扩展 SEI 消息建立 H.264 体系内部的误码信息反馈通道; 通过分段逐次帧内编码实现误码扩散抑制。



1. 一种基于 H.264 的压缩视频传输误码消除方法，其特征在于，包含以下步骤，

发送端接收接收端反馈的误码信息，所述接收端用于统计所述误码信息并实行误码掩盖策略；

所述发送端根据所述误码信息获得丢失条带的位置信息，从所述丢失条带中分割出一段连续宏块组成新条带，对所述新条带进行帧内编码，并在下一帧时发送。

2. 根据权利要求 1 所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，还包括：所述接收端通过定义扩展补偿增强消息来承载所述误码信息，将所述误码信息反馈给所述发送端。

3. 根据权利要求 2 所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，所述扩展补偿增强消息的载荷类型定义为用于承载所述误码信息。

4. 根据权利要求 1 所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，所述分割出一段连续宏块的大小满足以下条件：

该段宏块进行帧内编码后，本帧的数据率在 H.264 数据率控制范围内。

5. 根据权利要求 4 所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，所述从所述丢失条带中分割出一段连续宏块组成新条带还包含以下子步骤：

从所述丢失条带所包含的第一个宏块开始分割出一段连续宏块，组成新条带；如果所述丢失条带中当前包含的宏块数目大于或者等于预先设定的参数 P，则该连续宏块段包含的宏块数目是 P，否则该连续宏块段包含所述丢失条带中的所有宏块；在划分出该连续宏块段后，如果原来的丢失条带中还

有剩余的宏块，则这些剩余宏块仍属于所述丢失条带，此后进入所述对所述新条带进行帧内编码，并在下一帧时发送的步骤；

其中所述预先设定的参数 P 满足以下条件：

在进行 P 个宏块的帧内编码后，本帧的数据率保持在 H.264 数据率控制范围内。

6. 根据权利要求 1 至 3 中任一项所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，所述接收端统计所述误码信息并实行误码掩盖策略包括：

所述接收端检测误码，并统计误码信息；

所述接收端在检测到误码发生后，进行压缩视频信息重同步；

所述接收端根据所述误码信息实行所述误码掩盖策略。

7. 根据权利要求 6 所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，所述接收端根据网络抽象层单元序号的不连续情况来检测并统计误码信息。

8. 根据权利要求 7 所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，所述接收端根据所述网络抽象层单元序号的不连续情况获得丢失条带的位置信息，该位置信息包含所述丢失条带所在的帧序列号和所述丢失条带在该帧内的位置。

9. 根据权利要求 8 所述的基于 H.264 的压缩视频传输误码消除方法，其特征在于，所述误码掩盖策略包含以下步骤：

所述接收端用所述丢失条带所在帧的前一帧的对应位置上的相应条带，来替代该丢失条带。

基于 H.264 的压缩视频传输误码消除方法

技术领域

本发明涉及压缩视频传输误码消除方法，特别涉及基于 H.264 的压缩视频传输误码掩盖方法和误码扩散抑制方法。

背景技术

国际电信联盟电信标准部（International Telecommunication Union Telecommunication Standardization Sector，简称“ITU-T”）联合国际标准化组织（International Organization for Standardization，简称“ISO”）和国际电工委员会（International Electrotechnical Commission，简称“IEC”）的运动图像专家组（Moving Picture Expert Group，简称“MPEG”）制定的 H.264 压缩视频压缩编码标准，目前已经逐渐成为多媒体通信中的主流标准。大量采用 H.264 标准的多媒体实时通信产品，比如会议电视、可视电话、第三代（3rd Generation，简称“3G”）移动通信终端，以及网络流媒体产品先后问世。是否支持 H.264 已经成为这个市场领域中决定产品竞争力的关键因素。尤其是随着 3G 移动通信系统的出现和网际协议（Internet Protocol，简称“IP”）网络的迅速发展，视频网络通信正逐步成为通信的主要业务之一。ITU-T 继制定了 H.261、H.263、H.263+ 等视频压缩标准后，于 2003 年正式发布了 H.264 标准，它同时也是 MPEG-4 第 10 部分的主要内容。H.264 标准的制定更加有效地提高了视频编码效率和网络适配性，随着 H.264 的推广和使用，IP 网络和移动无线网络的多媒体通信进入了一个飞跃发展的新阶段。

下面简单介绍 H.264 标准的信息构成及发送机制：H.264 标准采用分层模式，定义了视频编码层（Video Coding Layer，简称“VCL”）和网络抽象层

(Network Abstraction Layer , 简称“NAL”) , 后者专为网络传输设计, 能适应不同网络中的压缩视频传输, 进一步提高网络的“亲和性”。H.264 引入了面向 IP 包的编码机制, 有利于网络中的分组传输, 支持网络中压缩视频的流媒体传输; 具有较强的抗误码特性, 特别适应丢包率高、干扰严重的无线压缩视频传输的要求。H.264 的所有待传送数据, 包括图像数据及其他消息均封装为统一格式的包传送, 即网络抽象层单元 (NAL Unit, 简称“NALU”) 。每个 NALU 是一个一定语法元素的可变长字节字符串, 包括包含一个字节的头信息, 可用来表示数据类型, 以及若干整数字节的负荷数据。一个 NAL 单元可以携带一个编码片、各自类型数据分割或一个序列或图像参数集。为了加强数据可靠性, 每帧图像都被分为若干个条带 (Slice) , 每个 Slice 由一个 NALU 承载, Slice 又是由若干个更小的宏块组成, 即为最小的处理单元。一般的说, 前后帧对应位置的 Slice 相互关联, 不同位置的 Slice 相互独立, 这样可以避免 Slice 之间发生误码相互扩散。

H.264 数据包含非参考帧的纹理数据、序列参数、图像参数、补充增强消息 (Supplemental Enhancement Information, 简称“SEI”) 、参考帧纹理数据等。其中, SEI 消息是在 H.264 压缩视频的解码、显示及其它方面起辅助作用的消息的统称。现有技术定义了各类 SEI 消息, 同时保留了 SEI 预留消息, 为未来的各种可能应用留下了扩展余地。根据 H.264, SEI 消息并非在解码过程重构亮度和色度图像所必需的。符合 H.264 标准的解码器, 是不需要对于 SEI 作任何处理的。也就是说, 不是所有符合 H.264 基本要求的终端都能够处理 SEI 消息的, 但是对于不能处理 SEI 消息的终端, 发送 SEI 对于它是没有影响的, 它会简单地忽略掉它不能处理的 SEI 消息。按照 SEI 语法规则, 用户可以利用预留消息传送自定义消息, 实现功能扩展。

由于 H.264 使用多种高效编码算法, 使得压缩视频码流对信道误码的敏感度增加, 即使单个原发性错误也可能会造成恢复视频质量的急剧下降。比如在 IP 网络中, 虽然采用了很多的承载层的服务质量 (Quality of Service,

简称“QoS”)管理策略,但是不可避免地存在网络带宽波动,导致丢包、包延迟等问题频繁发生。这类问题产生的传输错误,称为删除错误(Erasure Error),与之不同的是传统电路交换网络上的随机位错误(Random Bit Error),相对于随机位错误,删除错误更难防止和纠正。在实际 H.264 压缩视频通信中,由于丢包等引起的删除错误导致图像质量退化是非常严重的,更甚于引起解码端系统的崩溃。这是由于 H.264 相对于其它压缩视频编码标准来说能力更强、效率更高、功能更丰富,反过来对于删除错误的承受能力也更低。因此,在基于 H.264 标准的压缩视频通信中,必须采用有效的抗丢包等删除错误的技术,并结合多种视频抗误码方法,来保证恢复图像的质量。

在典型的分组 IP 网络中,删除误码就是丢包错误。现有的抗丢包错误技术即误码消除技术有很多种,比如纠删编码(Erasure Codes),网络自动重传请求(Automatic Retransmission Request,简称“ARQ”),交织打包(Interleaving),误码掩盖(Error Concealment)等。这些方法按照出发点不同大体可以分为两类:(a)主动防错型:事先采取保护措施,比如引入冗余机制,尽量保证数据包不丢失或者确保接收端能够恢复少量丢失的数据;(b)错误补偿型:在发生误码情况下采取一定的补偿措施,比如在网络状况恶化严重情况下,丢包率非常高,主动防错方法失去效果,这时就需要对已经发生的误码进行补偿。

错误补偿的误码消除方法根据侧重点不同又分为误码掩盖和误码扩散抑制两种。其中,误码掩盖是侧重于补偿误码当前的影响,比如接收端当前视频帧或者 Slice 丢失,则图像无法正确显示,即采取一定措施进行补偿,使得对于用户产生的影响最小。而误码扩散抑制则是消除误码在空间和时间上扩散带来的后续影响,比如接收端收到的帧丢失或者部分丢失,由于该帧可能是后续帧的预测参考帧,其误码将会在时间域上扩散到后续帧中;或者由于 H.264 中可能存在的帧内预测,以及环路滤波,都可能使得该帧的误码通过空间预测扩散到该帧的其它位置。误码扩散抑制就是采用一定的措施,在

空间上要限制误码影响在有限区域内，在时间上要限制在有限时间内，避免视频通信失败，更甚于解码器系统工作紊乱和崩溃。

可见在误码环境下，误码扩散不仅使出错帧的恢复图像质量下降，而且可能会给后续帧造成不可恢复的损失，即使解码端使用了误码掩盖技术，也不能避免恢复图像质量的下降。另外，由于视频通信的实时性要求强，通常不采用 ARQ 方式重传发生错误的帧。

通过误码掩盖，即通过用发生误码部分在空间上和时间内相邻部分的正确数据进行简单代替或者复杂的预测，插值等掩盖错误部分的方法，可以解决补偿误码问题。这在接收端即可实现，无需发送端参与。而误码扩散问题则更加复杂，需要发送端和接收端配合起来采取适当的策略才能抑制和消除。

需要注意的是，误码掩盖也会导致误码扩散的产生。事实上，由于误码掩盖会造成编码端和解码端重构图像缓存内容不匹配，从而导致误码在时间域上的扩散。例如，当解码第 $n-1$ 帧有丢包时，解码端会使用第 $n-2$ 帧对应位置图像数据进行误码掩盖，而在发送端编码时并不知第 $n-1$ 帧有丢包，会使用正确的第 $n-1$ 帧图像来编码第 n 帧图像，而在接收端解码第 n 帧时却会用第 $n-2$ 帧代替第 $n-1$ 帧解码，由此引起误码扩散。

现有的误码消除方法都是独立的误码掩盖方法或者误码扩散抑制方法，都有很多种实现技术细节不同的方法。误码掩盖方法有时间域掩盖、空间域掩盖、时空联合掩盖等方法。误码扩散抑制又有帧内编码、标识、自适应帧内块刷新等方法。

时间域掩盖方法就是采用时间轴上相邻的帧的信息来推算丢失数据。推算的方法可以是：简单采用相邻帧相同位置的数据代替丢失数据；考虑运动预测因素，根据相邻帧数据进行运动预测。除此还有更加复杂的掩盖策略，但是计算量非常大。

空间域掩盖方法就是利用丢失数据区域的空间相邻区域来进行错误掩

盖。同样的有：简单用领域替代；基于数据融合的有多个空间相邻区域推算丢失数据，比如空间插值；代数反演法，把丢包过程用一个线性模型建模，其输入是丢包前数据，输出是正确接收到的数据，利用代数反演的方法，比如最小二乘法，从输出来反演输入，用反演结果来替代错误数据，这种方法计算量大。

时空联合掩盖方法则是联合使用空间域和时间域的误码掩盖。比如，根据丢失数据的特点和相邻时间数据和空间数据的情况，采用某种策略确定用空间域掩盖还是时间域掩盖更好，然后实施这种更好的掩盖策略。或者，融合空间数据和时间数据，共同进行掩盖。

基于帧内编码的误码扩散抑制方法是将受误码影响的宏块采用帧内编码，即利用运动矢量的前向依赖关系进行准确误码跟踪，并对受误码影响的宏块采用帧内编码，可以有效地防止误码扩散。首先给出由于运动补偿所引起的帧间依赖性；然后根据运动矢量前向依赖性和权重因子的相关性计算误码的“能量”，对“能量”最大的宏块使用帧内编码，从而防止误码扩散。

基于标识的误码扩散抑制方法是将受误码影响的宏块做标识，使得编码时避免采用标识过的宏块做参考帧，直接防止了扩散。该方法需要建立重发送端到接收端的反馈机制，接收端将丢失数据的信息反馈到发送端，编码端根据错误信息将同一块组中出错宏块以后的像素全部用一个特定的值来标识，在之后的若干帧的编码中不参考已标识的区域，避免了接收端的误码扩散。

基于自适应帧内块刷新策略的误码扩散抑制方法是基于编码端的“误码灵敏性尺度”来衡量每一个编码宏块对信道误码的易损性，然后进行自适应的帧内块刷新。该方法不需要反馈信道。编码端先初始化“误码灵敏性尺度”值：距离同步标志越远的宏块，对误码的敏感性越高；编码宏块的比特数越多，越容易受到误码破坏。在编码过程中，通过计算每个宏块“误码灵敏性

尺度”值的积累来更新这个尺度，然后根据 ESM 尺度选择宏块进行帧内编码。

在实际应用中，上述方案存在以下问题：上述误码掩盖方法只能暂时掩盖误码导致的失真，而且简单的方法产生的效果不好，复杂的方法计算量大，另外掩盖替代还会加重误码扩散；

上述误码扩散抑制方法实现机制都比较复杂，加重网络负担外，算法过于复杂不利于实时处理应用，另外还不能完全去除误码扩散的影响，导致恢复图像质量仍会下降。

造成这种情况的主要原因在于，独立的误码掩盖方法采用的替代机制会引起误码扩散；误码扩散抑制方法都需要复杂的机制或者额外的反馈信道，耗费系统处理资源和网络带宽资源。

发明内容

有鉴于此，本发明的主要目的在于提供一种基于 H.264 的压缩视频传输误码消除方法，使得可以简单高效地消除误码。

为实现上述目的，本发明提供了一种基于 H.264 的压缩视频传输误码消除方法，包含以下步骤，

发送端接收接收端反馈的误码信息，所述接收端用于统计所述误码信息并实行误码掩盖策略；所述发送端根据所述误码信息获得丢失条带的位置信息，从所述丢失条带中分割出一段连续宏块组成新条带，对所述新条带进行帧内编码，并在下一帧时发送。其中，所述接收端通过定义扩展补偿增强消息来承载所述误码信息，将所述误码信息反馈给所述发送端。

此外，所述扩展补偿增强消息的载荷类型定义为用于承载所述误码信息。

此外，所述分割出一段连续宏块的大小满足以下条件：

该段宏块进行帧内编码后，本帧的数据率在 H.264 数据率控制范围内。

此外，所述从所述丢失条带中分割出一段连续宏块组成新条带还包含以下子步骤：

从所述丢失条带所包含的第一个宏块开始分割出一段连续宏块，组成新条带；如果所述丢失条带中当前包含的宏块数目大于或者等于预先设定的参数 P，则该连续宏块段包含的宏块数目是 P，否则该连续宏块段包含所述丢失条带中的所有宏块；在划分出该连续宏块段后，如果原来的丢失条带中还有剩余的宏块，则这些剩余宏块仍属于所述丢失条带，此后进入所述对所述新条带进行帧内编码，并在下一帧时发送的步骤；

其中所述预先设定的参数 P 满足以下条件：

在进行 P 个宏块的帧内编码后，本帧的数据率保持在 H.264 数据率控制范围内。

此外，所述接收端统计所述误码信息并实行误码掩盖策略包括：

所述接收端检测误码，并统计误码信息；

所述接收端在检测到误码发生后，进行压缩视频信息重同步；

所述接收端根据所述误码信息实行所述误码掩盖策略。

此外，所述接收端根据网络抽象层单元序号的不连续情况来检测并统计

误码信息。

此外，所述接收端根据所述网络抽象层单元序号的不连续情况获得丢失条带的位置信息，该位置信息包含所述丢失条带所在的帧序列号和所述丢失条带在该帧内的位置。

此外，所述误码掩盖策略包含以下步骤：

所述接收端用所述丢失条带所在帧的前一帧的对应位置上的相应条带，来替代该丢失条带。

通过比较可以发现，本发明的技术方案与现有技术的主要区别在于，通过误码信息反馈机制结合误码掩盖和误码扩散抑制实现误码消除；

通过 NALU 序号及承载 Slice 的信息检测误码发生，并统计丢失数据位置等误码信息；

通过定义扩展 SEI 消息建立 H.264 体系内部的误码信息反馈通道；

通过分段逐次帧内编码实现误码扩散抑制。

这种技术方案上的区别，带来了较为明显的有益效果，即结合误码掩盖和误码扩散抑制，避免由误码掩盖引起的误码扩散，在简单复杂度前提下，达到理想的误码消除效果，提高压缩视频传输质量；

根据 NALU 序号来进行统计，不但能保证统计信息精确无误，而且节省系统资源；

用扩展 SEI 消息传递可以节省开销、简化机制，且保证系统兼容性；

分段逐次帧内编码方法实现起来简单，防止扩散效果明显，降低误码消除复杂度，保证压缩视频传输稳定性。

附图说明

图 1 是根据本发明的第一实施例的基于 H.264 的压缩视频传输误码消除方法流程图；

图 2 是根据本发明的第二实施例的基于分段逐次帧内编码的误码扩散抑制原理示意图；

图 3 是根据本发明的第三实施例的有无误码消除压缩视频图像效果对比示意图。

具体实施方式

为使本发明的目的、技术方案和优点更加清楚，下面将结合附图对本发明作进一步地详细描述。

本发明从误码掩盖和误码扩散抑制两个方面出发，结合接收端误码掩盖策略和发送端的误码扩散抑制策略，以实现既能尽量减少误码带来的视频质量损失又能避免误码引起扩散的目的。对于误码掩盖，采用简单替代方案即可达到以尽量低的复杂度实现补偿误码损失的效果；对于误码扩散抑制，通过 H.264 已有通道建立误码信息反馈机制，根据反馈实行帧内编码，以达到扩散消除效果，且不增加额外网络负担，确保压缩视频码流对误码问题的鲁棒性，也因此避免因误码掩盖引起的误码扩散。

本发明的基本思路是，在接收端通过对 NALU 序号的统计，发现丢失数据信息，如 Slice 的位置等，一方面采用高效算法对丢失数据进行简单替代以掩盖误码损失，另一方面将误码信息反馈给发送端。通过 H.264 的扩展 SEI

消息，建立从接收端到发送端的误码信息反馈通道。发送端获知误码信息后，立即采取分段逐次进行帧内编码的策略，将误码 Slice 分段刷新，以防止误码扩散。

图 1 示出了本发明的第一实施例的 H.264 压缩视频传输误码消除方法的流程。H.264 压缩视频通信过程中，发送端对待发送视频流数据进行编码，得到压缩视频码流，然后封装 NALU 并通过分组报文传送给接收端。接收端接收报文并进行解码，此时接收端需要判断压缩视频流数据是否有丢失，以进行后续的误码消除操作。误码消除流程大致分为掩盖、反馈、扩散消除三个大步骤。

在步骤 101 中，接收端根据 NALU 序号不连续情况来判断是否丢失数据，并统计丢失数据的信息，即误码信息。如前所述，NALU 是 H.264 压缩视频流数据传输的基本单位，每个 NALU 都有唯一连续的序号。因此，接收端根据接收到 NALU 序号是否有不连续，获知哪些 NALU 丢失了。从而可以实行针对丢失数据的误码掩盖策略。采用 NALU 序号来进行统计，不但能保证统计信息精确无误，而且直接利用现有数据信息，不需要额外的承载开销。

首先接收端通过识别接收到的 NALU 头信息获知序号，由序号的不连续检测误码发生，通过前面 NALU 得知中间缺失的 NALU 应该承载的压缩视频数据，对误码引起的数据丢失进行定位，比如丢失 NALU 的前一个 NALU 承载的是第 N 帧的第 1 个 Slice，则按传送顺序可以推断丢失 NALU 所承载的 Slice 的位置，应该是本帧的后一个 Slice。

接着接收端需要进行压缩视频信息的重同步，由于 H.264 压缩视频码流连续传输过程中，接收端与数据流需要同步，然后才能正确接收，一旦数据流有不连续之后，接收端需要重新进行同步，通过找到不连续处之后的下一个 NALU 头信息来完成解码器的重同步。这一过程，接收端也需要通过下一个 NALU 的序号来判断中间丢失的 NALU 个数及其位置信息。

之后，接收端需要进行误码掩盖，丢失数据的 NALU 被整个丢弃，因此该 NALU 所承载整个 Slice 丢失，误码掩盖策略的一种方式就是简单替代，用时间域或者空间域相邻的数据代替丢失的数据，比如采用丢失数据所在帧的前一帧对应位置的 Slice 恢复图像数据进行掩盖。

本发明的第一实施例中采用简单实用、计算效率高的接收端误码掩盖方法，而不单独追求掩盖的效果最好，而是综合考虑效果代价比，即在简单复杂度的前提下，实现理想的误码掩盖效果。熟悉本领域的技术人员可以理解，接收端在获知误码信息后，也可以采用掩盖效果较好、计算较复杂的误码掩盖方法，以尽力减少用户视频效果损失，而不影响本发明的实质和范围。

在步骤 102 中，接收端在获得误码信息后，将其反馈给发送端。反馈误码信息需要一条反馈通道，为了减少网络负担、简化实现机制，本发明的第一实施例中采用现有的 H.264 通信机制，定义扩展 SEI 消息，用于承载误码信息建立反馈，以便发送端结合误码信息防止误码扩散。事实上，结合误码信息反馈机制和发送端的误码扩散抑制策略，才能避免因前面接收端实行的误码掩盖策略导致的误码扩散。

利用 H.264 的扩展 SEI 消息提供一种从接收端到发送端的信息反馈机制，使得发送端可以及时了解哪些 NALU 丢失了，这样可以及时进行有效的误码扩散抑制，防止因这些丢失的数据引起以后的误码扩散。事实上，如果误码扩散抑制方法缺乏来自接收端的及时反馈，只在发送端单方面的进行预测和防止扩散，不但其效果会受到影响，而且往往计算复杂度很高。

在 H.264 体系内部建立信息反馈机制的好处在于节约网络带宽开销、节省系统处理资源，且不影响互通性。下面介绍如何定义扩展 SEI 消息。如前所述 SEI 消息也由 H.264 码流的基本单位 NALU 所承载，每个 SEI 域包含一个或多个 SEI 消息，而 SEI 消息又由 SEI 头信息和 SEI 有效载荷组成。SEI 头信息包括两个码字：载荷类型和载荷大小。其中载荷类型的长度不一定，

比如类型在 0 到 255 之间时用一个字节表示，当类型在 256 到 511 之间时用两个字节 0xFF00 到 0xFFFE 表示，依次类推，这样用户可以自定义任意多种载荷类型。在现有 H.264 标准中，类型 0 到类型 18 标准中已定义为特定的信息，如缓存周期、图像定时等。由此可见 H.264 中定义的 SEI 域可根据需求存放足够多的用户自定义信息。在本发明的第一实施例中，在预留的 SEI 载荷类型中定义一种用于承载统计信息的扩展 SEI 消息。

由于 SEI 消息是附属消息，有无 SEI 消息不影响正常的视频通信，因此本发明给出的扩展 SEI 消息并不影响现有的视频流通信，具有通用性。即如果通信双方终端都支持本发明方案，则可以利用 SEI 消息传递丢包统计数据，从而实施不同能力等级的自适应保护；如果有一方不支持，也不会影响正常的通信。可见，自定义的扩展 SEI 消息不会影响基于 H.264 压缩视频通信系统的兼容性。另外，采用 SEI 消息来传递丢包统计数据的另外一个好处是节省开销，SEI 是 H.264 码流的一部分，利用 H.264 码流本身来承载丢包统计数据，不需要开辟和维护额外的信道，传输高效、实现简单。

熟悉本领域的技术人员可以理解，接收端到发送端的误码信息反馈通道也可以通过定义通信协议创建专用的通道，或者通过基于 H.264 的其他预留通道建立，也能实现结合误码掩盖和误码扩散抑制的发明目的，而不影响本发明的实质和范围。

在步骤 103 中，发送端即开始根据反馈的误码信息进行误码扩散抑制。联合误码信息的误码扩散抑制方法，其效果要比现有的无反馈的误码扩散抑制好。利用误码信息，比如丢失 Slice 的位置，发送端可以有目的的针对所丢失 Slice 采取防止措施，比如在以后的编码中避免以丢失 Slice 作为参考帧，这样可以尽量缩短接收端解码时对该 Slice 的依赖。

由于 H.264 编码是基于 Slice 的，即前后帧的同一 Slice 的数据是具有参考关联的，后续帧的同一 Slice 数据是通过前面帧的 Slice 预测编码的，因此

误码扩散也将限定在同一 Slice 内部。本发明的第二实施例中，采用分段逐次进行帧内编码的策略，即在发送误码之后，对以后帧的该 Slice 区域分段分割为新的 Slice，比如划分处 P 个宏块作为一个新 Slice，然后对其采用帧内编码，以消除该 Slice 对前面丢失的 Slice 的参考或依赖。由于 H.264 压缩视频实时传输系统为了保证传输质量，采用数据率控制方案来限制每帧数据的波动，使得每帧数据量均衡，提高压缩视频传输的稳定性。因此，在每帧中一次进行帧内编码的数据量即宏块数目不能太多，否则将会超过 H.264 数据率控制范围。

熟悉本领域的技术人员可以理解，发送端在收到反馈误码信息后，可以通过其他方法，比如对丢失 Slice 做标记，避免以后编码时对其参考等，也能实现结合误码掩盖和误码扩散抑制的发明目的，而不影响本发明的实质和范围。

图 2 示出了本发明的第二实施例中分段逐次帧内编码的误码扩散抑制的原理。当接收端出现无法恢复的丢包错误后，检测并反馈误码信息给发送端，即丢失数据的 Slice 所在帧及帧内位置信息通过扩展的 SEI 消息发回给发送端。发送端从 SEI 消息中提取丢失的 slice 位置信息，比如图 2 中的每帧划分为三个 Slice，即 Slice#0、Slice#1、Slice#2，而第 n 帧的 Slice#1 在传输中丢失，之后需要执行分段逐次帧内编码。

首先，在第 n 帧中，编码端对 Slice#1 按宏块扫描顺序，从起始位置开始分割 P 个宏块组成新的 Slice#3，剩余宏块仍然为 Slice#1，此时有四个 Slice，其中对新的 Slice#3 进行帧内编码。

接着，在第 n+1 帧中，上一步中分割新组成的 Slice#3 在帧内编码之后，作为 Slice#3 发送出去，而其他 Slice 仍然按照常规编码。

此后，需要判断 Slice#1 中是否还剩余宏块，如果还有没有分割的，则返回第一步在下一帧中继续将 Slice#1 剩余宏块分段组成新的帧，实行帧内编

码并发送，直到所有宏块处理完毕。这里存在一个 Slice#1 中原来宏块数量不是 P 的整数倍的问题，如果不是，那么最后一次剩余的宏块数目是小于 P 的。

上面每次划分的宏块个数 P 应该满足以下条件，尽量大以避免分割次数、减少处理时延、缩短影响范围，但是需要满足前述 H.264 数据率控制范围。每次划分的宏块个数可以不一样，但最后一次划分的宏块数将使得丢失 Slice 中的所有宏块都处理完毕。

比如说压缩视频流数据的一帧由 396 个宏块组成，初始划分每 64 个宏块为一个 Slice，即 0 - 63 宏块为 Slice # 0，64 - 127 宏块为 Slice # 1，128 - 191 宏块为 Slice # 2,依此类推。而根据数据率计算确定合适分段数值 P 为 8 个宏块一段。则第 n 帧中丢失 Slice # 1 后，Slice # 1 的 64 个宏块应该进行分段逐次帧内编码，首先在第 n+1 帧中选前 8 个宏块进行帧内编码组成为 Slice # k，这样在第 n+2 帧中 Slice # k 即可采用常规预测编码，而接着的 8 个宏块再进行帧内编码组成 Slice # k+1，依次直到第 n+8 帧时最后剩余为 8 个宏块进行帧内编码组成 Slice # k+7，才完成分段逐次帧内编码的误码扩散方法流程。上述 k 是一个整数。

最后，从图像和数据上比较一下使用本发明的方法前后的实验结果。

图 3 给出了根据上述方式进行实验数据测试时产生的误码消除图像效果，可见采用本发明的误码掩盖和误码扩散抑制联合的方法后，得到的视频图像效果非常好。

下面的两个表中从数据上比较了使用误码掩盖和误码扩散抑制算法前后的恢复图像客观质量。这里使用了两个国际通用的测量图像处理压缩编码质量的测试视频序列——Foreman 图像序列和 Container 图像序列。通过峰值信噪比（Peak Signal-to-Noise Ratio，简称“PSNR”）进行比较，PSNR 是度量客观质量的最重要标准，数值越高越好。在下面两个表中，PSNR1 是未使用本发明的防误码扩散和误码掩盖时的 PSNR，PSNR2 是使用本发明的防误

码扩散和误码掩盖后的 PSNR。

Foreman 图像序列

丢包率:	0%	2%	4%	6%	8%	10%	12%	14%	16%
PSNR1	36.56	26.69	23.80	21.81	19.32	20.01	18.59	18.74	17.45
PSNR2	36.56	35.22	34.08	33.25	31.94	31.27	30.32	29.63	29.34

Container 图像序列

丢包率:	0%	2%	4%	6%	8%	10%	12%	14%	16%
PSNR1	35.97	34.61	32.17	30.84	29.91	28.87	27.71	26.58	25.51
PSNR2	35.97	35.37	34.82	34.65	34.10	33.19	32.33	31.85	31.38

虽然通过参照本发明的某些优选实施例，已经对本发明进行了图示和描述，但本领域的普通技术人员应该明白，可以在形式上和细节上对其作各种改变，而不偏离本发明的精神和范围。

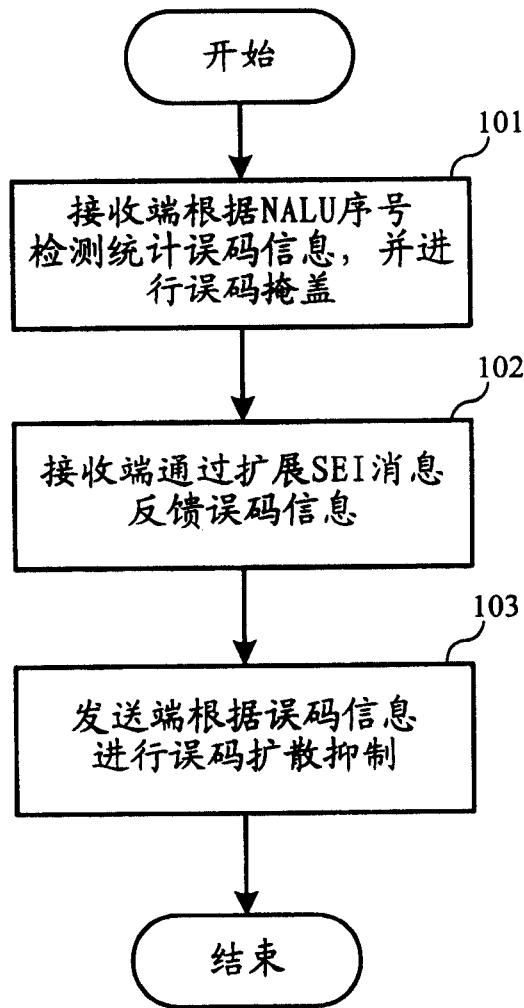
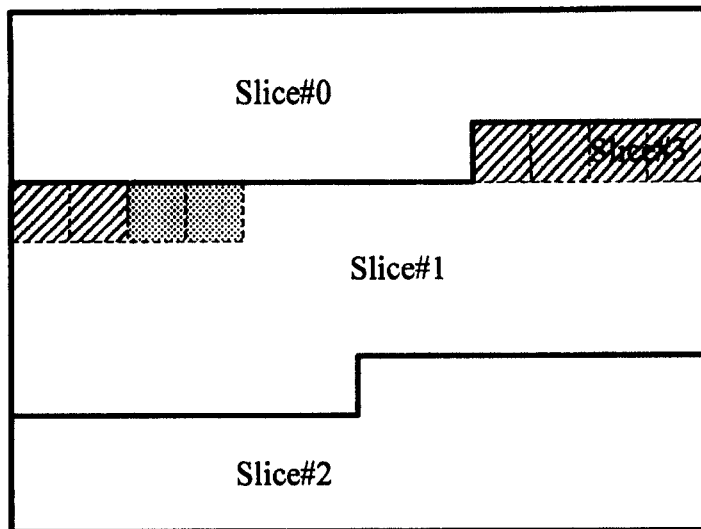


图 1



- ▨ 经过帧内刷新的宏块
- ▤ 未经过帧内刷新的宏块

图 2



无误码消除的压缩视频图像

有误码消除的压缩视频图像

图 3